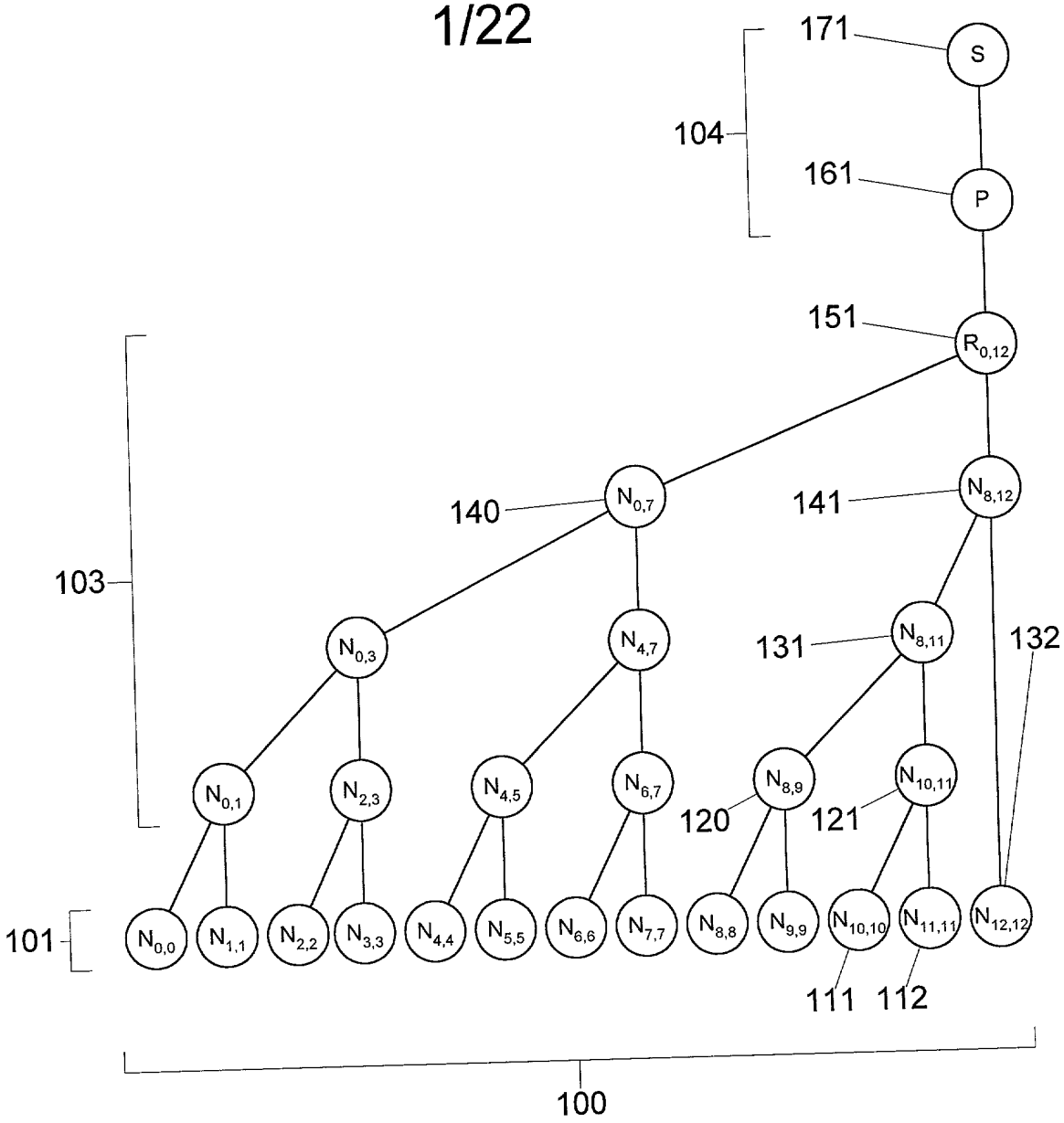

171
S
104
161
P
151
$R_{0,12}$
140
$N_{0,7}$
141
$N_{8,12}$
103
$N_{0,3}$
$N_{4,7}$
131
$N_{8,11}$
132
$N_{0,1}$
$N_{2,3}$
$N_{4,5}$
$N_{6,7}$
120
$N_{8,9}$
121
$N_{10,11}$
101
$N_{0,0}$
$N_{1,1}$
$N_{2,2}$
$N_{3,3}$
$N_{4,4}$
$N_{5,5}$
$N_{6,6}$
$N_{7,7}$
$N_{8,8}$
$N_{9,9}$
$N_{10,10}$
$N_{11,11}$
$N_{12,12}$
111
112
100

Figure 1

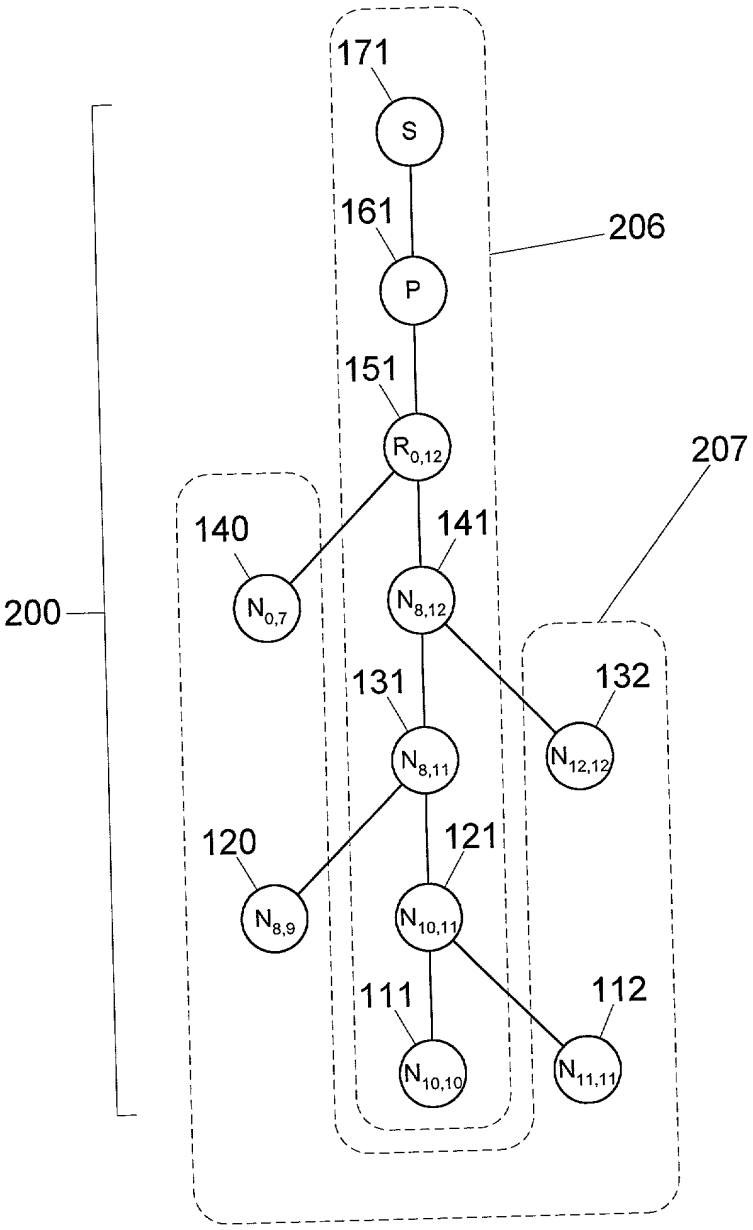

171
S
161
P
206
151
$R_{0,12}$
207
140
141
200
$N_{0,7}$
$N_{8,12}$
131
132
$N_{8,11}$
$N_{12,12}$
120
121
$N_{8,9}$
$N_{10,11}$
111
112
$N_{10,10}$
$N_{11,11}$

Figure 2


<signature
value=". . ."
pad=". . ."
size=13
>
<right
value=". . ."
/>
<left
value=". . ."
/>
<right
value=". . ."
/>
<left
value=". . ."
/>
</signature>
371
372
373
371T
312
312T
320
320T
307T
332
332T
340
340T
379T
300T

Figure 3

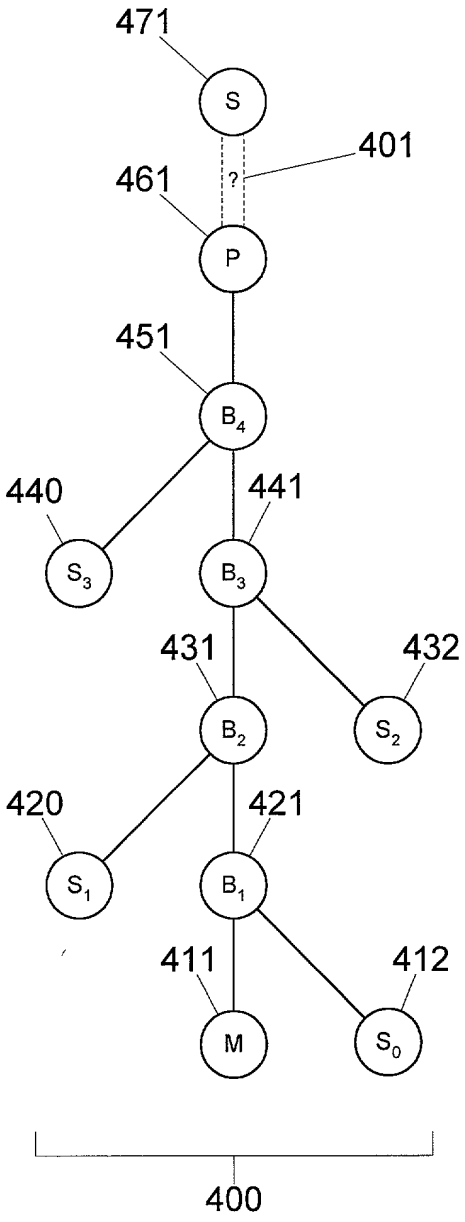

471
S
401
461
?
P
451
B4
440
441
S3
B3
431
432
B2
S2
420
421
S1
B1
411
412
M
S0
400

Figure 4

5.1 $\mathbf{N}$: the natural numbers $0, 1, 2, 3, \ldots$

5.2 L : a security parameter for cryptographic hash functions
5.3 $\mathbf{H} = \Sigma^L$ = set of all bit strings of length L
5.4 Σ^* = set of all bit strings of any length
5.5 $F : \Sigma^* \rightarrow \mathbf{H}$: a cryptographic hash function of output length L

5.6 P : a security parameter for a public key signature scheme
5.7 $\mathbf{J} = \Sigma^P$ = set of all bit strings of length P
5.8 $\mathbf{K}_E$: the space of private keys of the signature scheme
5.9 $\mathbf{K}_D$: the space of public keys of the signature scheme
5.10 $S_{EG} : \mathbf{K}_E \times \mathbf{J} \rightarrow \mathbf{J}$: the signing operation of a private key
5.11 $V_{EG} : \mathbf{K}_D \times \mathbf{J} \times \mathbf{J} \rightarrow \mathbf{B}$: the verification predicate of a public key

5.12 $\mathbf{H}^+$: positive length sequences of values in $\mathbf{H}$
5.13 $\# : \mathbf{H}^+ \rightarrow \mathbf{N}$: the length of a sequence
5.14 $\overline{M} = \{M_i\} \in \mathbf{H}^+$: a sequence of messages
5.15 $n = \# \overline{M}$: the number of input messages; also the size of a hash tree

5.16 Base10 : $\mathbf{N} \rightarrow \Sigma^*$: represents a number as an ASCII base-10 integer
5.17 Base64 : $\mathbf{N} \rightarrow \Sigma^*$: represents a number as an ASCII base-64 integer
5.18 quote : a constant, the ASCII quote character
5.19 $+ : \Sigma^* \rightarrow \Sigma^*$: string concatenation

5.20 $\mathbf{B} = \{true, false\}$: boolean values

Figure 5

6.1 $\mathbf{T}_n \subset \mathbf{N}\times\mathbf{N}$: the set of positions in an ordered tree of size n

6.2 $K:\mathbf{N}\times\mathbf{N}\to\mathbf{N}:\langle i,j\rangle\mapsto\min\left\{k\,\middle|\, j-i<2^k\right\}$

6.3 $\mathbf{T}_n=\left\{\langle i,j\rangle\,\middle|\, 0\le i\le j\le n-1\wedge\left(2^{K\langle i,j\rangle}\,\middle|\, i\right)\wedge j=\min\left(i+2^{K\langle i,j\rangle}-1,n-1\right)\right\}$

6.4 $\mathbf{T}_n^+=\left\{\langle i,j\rangle\,\middle|\,\langle i,j\rangle\in\mathbf{T}_n\wedge i\neq j\right\}$: the parent nodes in the tree

6.5 $\mathbf{T}_n^-=\left\{\langle i,j\rangle\,\middle|\,\langle i,j\rangle\in\mathbf{T}_n\wedge\langle i,j\rangle\neq\langle 0,n-1\rangle\right\}$: the child nodes in the tree

6.6 $\mathrm{Left}_n:\mathbf{T}_n^+\to\mathbf{T}_n^-:\langle i,j\rangle\mapsto\left\langle i,i+2^{K\langle i,j\rangle-1}-1\right\rangle$

6.7 $\mathrm{Right}_n:\mathbf{T}_n^+\to\mathbf{T}_n^-:\langle i,j\rangle\mapsto\left\langle i+2^{K\langle i,j\rangle-1},j\right\rangle$

6.8 $L:\mathbf{N}\times\mathbf{N}\to\mathbf{N}:\langle n,i\rangle\mapsto L'\left(n,i,\langle 0,n-1\rangle\right)$

6.9 $$L':\mathbf{N}\times\mathbf{N}\times\mathbf{T}_n\to\mathbf{N}:\langle n,i,t\rangle\mapsto\begin{cases}1+L'\left(n,i,\mathrm{Left}_n(t)\right) & \text{when } i\in\mathrm{Left}_n(t)\\ 1+L'\left(n,i,\mathrm{Right}_n(t)\right) & \text{when } i\in\mathrm{Right}_n(t)\\ 0 & \text{when } t=\langle i,i\rangle\end{cases}$$

6.10 $\mathrm{ordinary}_n:\mathbf{N}\times\mathbf{N}\to\mathbf{B}:\langle i,j\rangle\mapsto\left(\langle i,j\rangle\in\mathbf{T}_n\right)\wedge\left(j=i+2^{K\langle i,j\rangle}-1\right)$

6.11 $$\left[\begin{array}{c}\text{"ordinary nodes are stable"}\Leftrightarrow\\ \forall t\in\mathbf{N}\times\mathbf{N};m,n\in\mathbf{N}:\left(\mathrm{ordinary}_n(t)\wedge m>n\right)\to\left(\mathrm{ordinary}_m(t)\right)\end{array}\right.$$

Figure 6

7.1 $\mathbf{VT}_n = \{N : \mathbf{T}_n \to \mathbf{H}\}$: set of valued trees

7.2 $$\left[\begin{array}{c} \text{"}N \in \mathbf{VT}_n \text{ is a well-constructed hash tree"} \Leftrightarrow \\ \forall t,u,v \in \mathbf{T}_n : u = \mathrm{Left}_n(t) \wedge v = \mathrm{Right}_n(t) \to N(t) = \mathrm{NodeHash}_n(t, N(u), N(v)) \end{array}\right.$$

7.3 $\mathrm{NodeHash}_n : \mathbf{T}_n \times \mathbf{H} \times \mathbf{H} \to \mathbf{H} : \langle\langle i,j\rangle, x, y\rangle \mapsto F \circ \mathrm{NodeFormat}_n(\langle i,j\rangle, x, y)$

7.4 $\mathrm{NodeFormat}_n : \mathbf{T}_n \times \mathbf{H} \times \mathbf{H} \to \Sigma^*$: a position-dependent formatting function

7.5 $N_{\overline{M}} \in \mathbf{VT}_{\#\overline{M}}$: hash tree generated from leaves $\overline{M}$

7.6 $$N_{\overline{M}} : t \mapsto \begin{cases} \mathrm{NodeHash}_{\#\overline{M}}(t, N_{\overline{M}} \circ \mathrm{Left}_{\#\overline{M}}(t), N_{\overline{M}} \circ \mathrm{Right}_{\#\overline{M}}(t)) & \text{when } t \in \mathbf{T}^+_{\#\overline{M}} \\ M_i & \text{when } t = \langle i,i\rangle \end{cases}$$

7.7 $$\left[\begin{array}{c} \text{"ordinary nodes have stable values"} \Leftrightarrow \\ \forall t \in \mathbf{T}_n, \overline{M}, \overline{P} \in \mathbf{H}^+ : (\mathrm{ordinary}_{\#\overline{M}}(t) \wedge \overline{M} \subset \overline{P}) \to (N_{\overline{M}}(t) = N_{\overline{P}}(t)) \end{array}\right.$$

7.8 $$\left[\begin{array}{c} \text{"hash tree } N \in \mathbf{VT}_n \text{ is valid for message sequence } \overline{M}\text{"} \Leftrightarrow \\ N = N_{\overline{M}} \end{array}\right.$$

Figure 7

8.1 $$\mathrm{NodeFormat}_n : \langle\langle i,j\rangle, x, y\rangle \mapsto \begin{cases} \mathrm{RootFormat}(n,x,y) & \text{when } \langle i,j\rangle = \langle 0, n-1\rangle \\ \mathrm{ParentFormat}(x,y) & \text{when } \langle i,j\rangle \in \mathbf{T}_n^+ \cap \mathbf{T}_n^- \end{cases}$$

8.2
$$\begin{aligned} \mathrm{RootFormat}(n,x,y) = {} & \text{"<root"} \\ & + \text{"size="} + \mathrm{Base10}(n) \\ & + \text{"left="} + \mathrm{quote} + \mathrm{Base64}(x) + \mathrm{quote} \\ & + \text{"right="} + \mathrm{quote} + \mathrm{Base64}(y) + \mathrm{quote} \\ & + \text{"/>"} \end{aligned}$$

8.3
$$\begin{aligned} \mathrm{ParentFormat}(x,y) = {} & \text{"<node"} \\ & + \text{"left="} + \mathrm{quote} + \mathrm{Base64}(x) + \mathrm{quote} \\ & + \text{"right="} + \mathrm{quote} + \mathrm{Base64}(y) + \mathrm{quote} \\ & + \text{"/>"} \end{aligned}$$

Figure 8

9.1 $\mathbf{U}_n = \mathbf{T}_n \cup \{P, S\}$: node set for combined signature tree

9.2 $R = \langle 0, n-1\rangle \in \mathbf{T}_n \subset \mathbf{U}_n$: root node of hash tree

9.3
$$\begin{cases} \mathrm{parent}(R) = P \\ \mathrm{parent}(P) = S \end{cases}$$

Figure 9

10.1 $\mathbf{VU}_n = \{N : \mathbf{U}_n \to \Sigma^*\}$: set of valued trees

10.2
$$\left[\begin{array}{l} \text{"}N \in \mathbf{VU}_n \text{ is a well-constructed combined signature tree"} \Leftrightarrow \\ \qquad N|_{\mathbf{T}_n} \text{ is a well-constructed hash tree"} \\ \qquad \wedge\, N(R) = \text{PadStrip}(N(P)) \\ \qquad \wedge\, \exists d \in \mathbf{K}_D : V_{EG}(d, N(S), N(P)) \end{array}\right.$$

10.3 $N_{e,\overline{M},z} \in \mathbf{VU}_{\#\overline{M}}$: combined signature tree generated by $e \in \mathbf{K}_E, \overline{M} \in \mathbf{H}^+, z \in \Sigma^{P-L}$

10.4
$$N_{e,\overline{M},z} : u \mapsto \begin{cases} S_{EG}(e, N_{e,\overline{M},z}(P)) & \text{if } u = S \\ \text{PadFormat}(N_{e,\overline{M},z}(R), z) & \text{if } u = P \\ N_{\overline{M}}(u) & \text{if } u \in \mathbf{T}_{\#\overline{M}} \end{cases}$$

10.5
$$\left[\begin{array}{l} \text{"combined signature tree } N \in \mathbf{VU}_n \text{ is valid for } d \in \mathbf{K}_D, \overline{M} \in \mathbf{H}^+\text{"} \Leftrightarrow \\ \qquad \text{"}N \text{ is a well-constructed combined signature tree"} \\ \qquad \wedge\, V_{EG}(d, N(S), N(P)) \\ \qquad \wedge\, N(R) = N_{\overline{M}}\langle 0, \#\overline{M} - 1\rangle \end{array}\right.$$

Figure 10

11.1 $\text{PadFormat} : \mathbf{H} \times \Sigma^{P-L} \to \mathbf{J}$: hash value padding function

11.2 $\text{PadFormat} : \langle h, z\rangle \mapsto h + z$

11.3 $\text{PadStrip} : \mathbf{J} \to \mathbf{H}$: pad stripping

11.4 $\text{PadStrip} : j \mapsto j[0, L-1]$

11.5 $\text{Padding} : \mathbf{J} \to \Sigma^{P-L}$: pad extraction

11.6 $\text{Padding} : j \mapsto j[L, P-1]$

11.7 "inverse on first parameter" $\Leftrightarrow \forall h, z \in \Sigma^* : h = \text{PadStrip}(\text{PadFormat}(h, z))$

11.8 "inverse on second parameter" $\Leftrightarrow \forall h, z \in \Sigma^* : z = \text{Padding}(\text{PadFormat}(h, z))$

Figure 11

12.1 $L = L(n,i);\ k,i \in [0,n-1]$: sequence parameters

12.2 $B_{n,i,0}, B_{n,i,1}, \ldots, B_{n,i,L+1} \in T_n$: branch nodes

12.3 $S_{n,i,0}, S_{n,i,1}, \ldots, S_{n,i,L} \in T_n$: sibling nodes

12.4 $side_{n,i,0}, side_{n,i,1}, \ldots, side_{n,i,L} \in \{\text{"left"}, \text{"right"}\}$: sibling positions

12.5 $B_{n,i,L+1} = \langle 0, n-1 \rangle$: root node

12.6 $$side_{n,i,k} = \begin{cases} \text{"left"} & \text{when } i \in \text{Right}_n\left(B_{n,i,k+1}\right) \\ \text{"right"} & \text{when } i \in \text{Left}_n\left(B_{n,i,k+1}\right) \end{cases}$$

12.7 $$S_{n,i,k} = \begin{cases} \text{Left}_n\left(B_{n,i,k+1}\right) & \text{when } side_{n,i,k} = \text{"left"} \\ \text{Right}_n\left(B_{n,i,k+1}\right) & \text{when } side_{n,i,k} = \text{"right"} \end{cases}$$

12.8 $$B_{n,i,k} = \begin{cases} \text{Right}_n\left(B_{n,i,k+1}\right) & \text{when } side_{n,i,k} = \text{"left"} \\ \text{Left}_n\left(B_{n,i,k+1}\right) & \text{when } side_{n,i,k} = \text{"right"} \end{cases}$$

Figure 12

13.1 $\text{ExtractedSignature} : \mathbf{K}_E \times \mathbf{H}^+ \times \Sigma^{P-L} \times \mathbf{N} \to \Sigma^*$

13.2 $\text{SiblingSeq} : \mathbf{H}^+ \times \mathbf{N} \times \mathbf{N} \to \Sigma^*$

13.3 $$\text{ExtractedSignature} : \langle e, \overline{M}, z, t \rangle \mapsto \left[\begin{array}{l} \text{SignTagStart}\left(N_{e,\overline{M},z}(S), z, \#\overline{M}\right) \\ + \text{SiblingSeq}\left(\overline{M}, t, 0\right) \\ + \text{SignTagEnd}() \end{array} \right]$$

13.4 $$\text{SiblingSeq} : \langle \overline{M}, t, k \rangle \mapsto \begin{cases} \left[\begin{array}{l} \text{SibTag}\left(side_{\#\overline{M},t,k}, N_{\overline{M}}\left(S_{\#\overline{M},t,k}\right)\right) \\ \quad + \text{SiblingSeq}\left(\overline{M}, t, k+1\right) \end{array} \right] & \text{when } k < L\left(0, \#\overline{M} - 1\right) \\ "" & \text{when } k = L\left(0, \#\overline{M} - 1\right) \end{cases}$$

13.5 $$\text{SignTagStart} : \mathbf{J} \times \Sigma^* \times \mathbf{N} \to \Sigma^* : \langle j, z, n \rangle \mapsto \left[\begin{array}{l} "< \text{signature}" \\ + " \text{value} =" + \text{quote} + \text{Base64}(j) + \text{quote} \\ + " \text{pad} =" + \text{quote} + \text{Base64}(z) + \text{quote} \\ + " \text{size} =" + \text{Base10}(n) \\ + ">" \end{array} \right]$$

13.6 $\text{SignTagEnd} : \quad \to \Sigma^* : \quad \mapsto "< /\text{signature} >"$

13.7 $$\text{SibTag} : \Sigma^* \times \mathbf{H} \to \Sigma^* : \langle s, v \rangle \mapsto \left[\begin{array}{l} "<" + s \\ + " \text{value} =" + \text{Base64}(v) \\ + "/>" \end{array} \right]$$

Figure 13

14.1 extracted signature $\Rightarrow$
$$\begin{cases} sig \in \mathbf{J} : \text{signature value} \\ z \in \Sigma^{P-L} : \text{padding bits} \\ n \in \mathbf{N} : \text{stated size of the tree} \\ L \in \mathbf{N} : \text{number of siblings on branch} \\ p_0 \dots p_{L-1} \in \{left, right\} : \text{position indicators} \\ s_0 \dots s_{L-1} \in \mathbf{H} : \text{sibling values} \end{cases}$$

14.2 $XS \in \mathbf{K}_D \times \mathbf{H} \times \mathbf{J} \times \Sigma^{P-L} \times \mathbf{N} \times \mathbf{N} \times \{left, right\}^+ \times \mathbf{H}^+$: extracted signature verification

14.3 $XS = \langle d, M, sig, z, n, L, \{p_0 \dots p_{L-1}\}, \{s_0 \dots s_{L-1}\} \rangle$

14.4 $\mathbf{W}_L = \{S_0 \dots S_{L-1}, B_0 \dots B_L, P\}$: nodes of the base tree of a verification tree

14.5
$$\begin{cases} \text{parent}(S_i) = B_{i+1} \\ \text{parent}(B_i) = B_{i+1}, \text{ when } i < L \\ \text{parent}(B_L) = P \end{cases} : \text{edges of the base tree of a verification tree}$$

14.6 $N_{XS} \in \mathbf{VW}_L = \{N : \mathbf{W}_L \to \Sigma^*\}$: verification tree

14.7
$$N_{XS} : w \mapsto \begin{cases} M & \text{when } w = B_0 \\ s_i & \text{when } w = S_i \\ \text{NodeHash}_n(P(B_i), N_{XS}(S_{i-1}), N_{XS}(B_{i-1})) & \text{when } w = B_i, 0 < i \le L, p_i = left \\ \text{NodeHash}_n(P(B_i), N_{XS}(B_{i-1}), N_{XS}(S_{i-1})) & \text{when } w = B_i, 0 < i \le L, p_i = right \\ \text{PadFormat}(N_{XS}(B_L), z) & \text{when } w = P \end{cases}$$

14.8
$$P_{XS} : \{B_0 \dots B_L\} \to T_n : w \mapsto \begin{cases} \text{Left}_n(B_{i+1}) & \text{when } w = B_i, i < L, p_i = right \\ \text{Right}_n(B_{i+1}) & \text{when } w = B_i, i < L, p_i = left \\ \langle 0, n-1 \rangle & \text{when } w = B_L \end{cases}$$

14.9
$$\text{"signature is valid for } \langle d, M \rangle\text{"} \leftrightarrow \begin{cases} \text{"signature is well-formed"} \\ \wedge\ P_{XS}(B_0) \text{ is defined} \\ \wedge\ \exists i : P_{XS}(B_0) = \langle i, i \rangle \\ \wedge\ V_{EG}(d, sig, N_{XS}(P)) \end{cases}$$

Figure 14

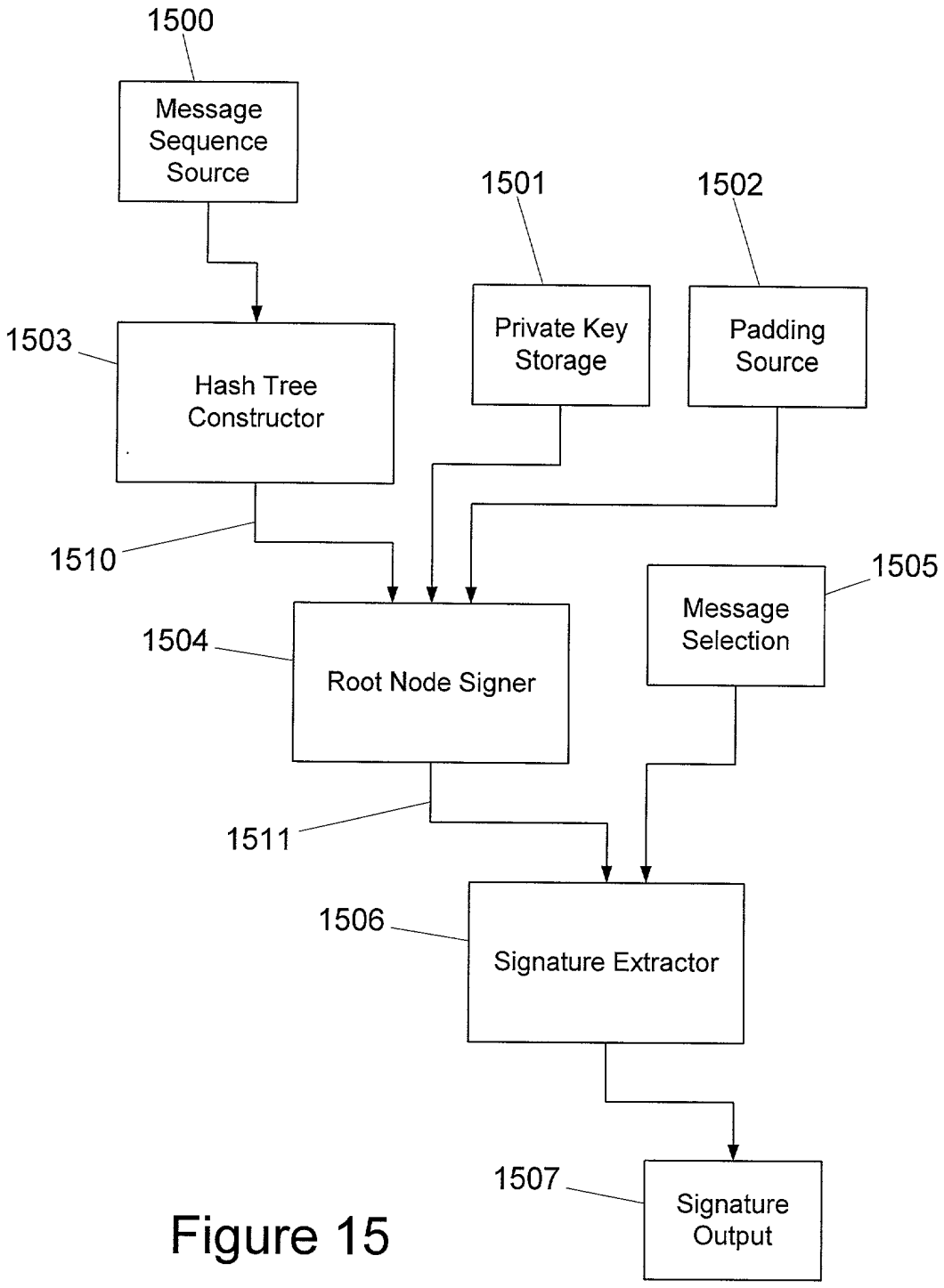

1500
Message Sequence Source
1501
1502
Private Key Storage
Padding Source
1503
Hash Tree Constructor
1510
1505
Message Selection
1504
Root Node Signer
1511
1506
Signature Extractor
1507
Signature Output

Figure 15

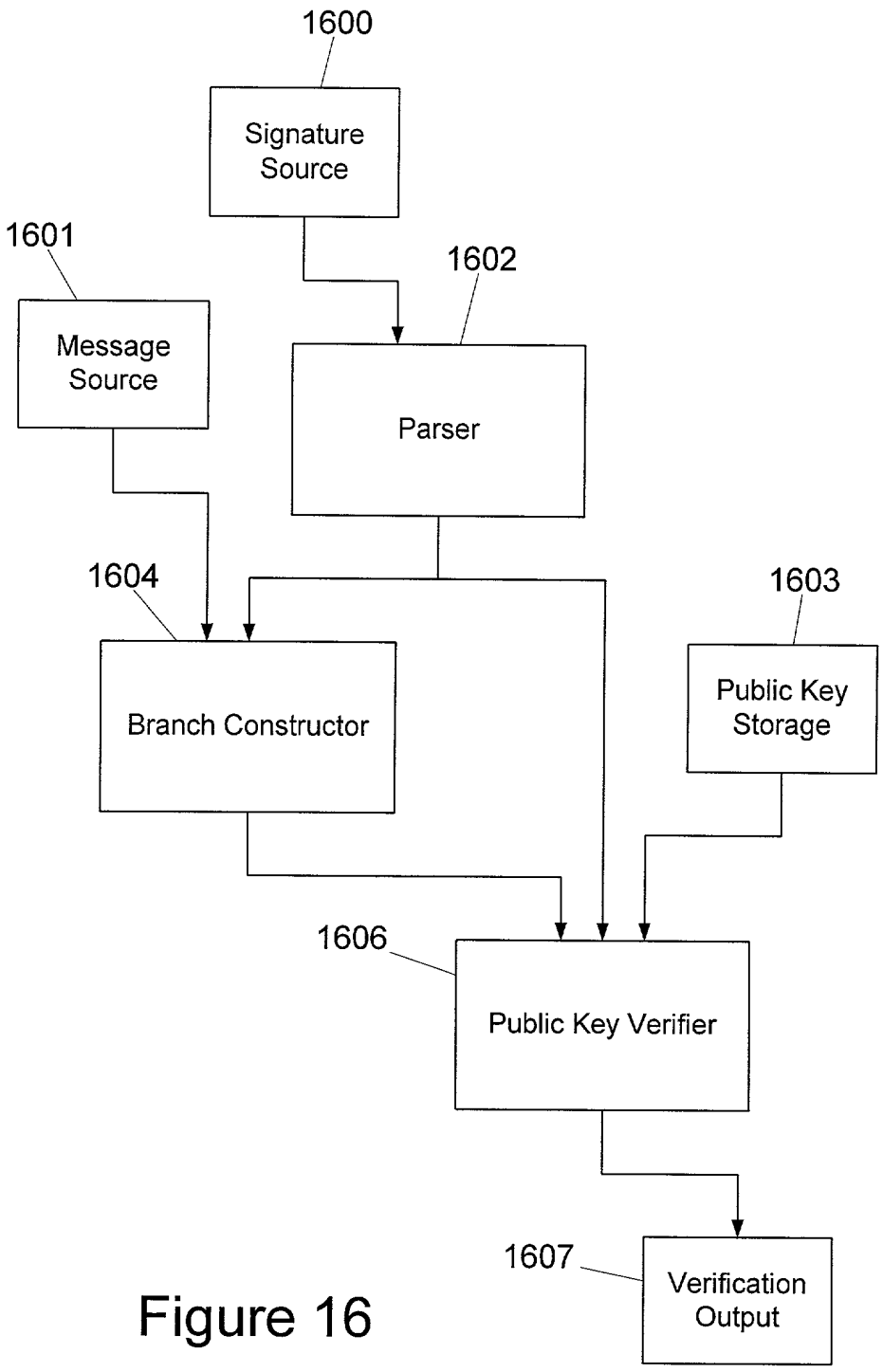

1600
Signature Source
1601
Message Source
1602
Parser
1604
Branch Constructor
1603
Public Key Storage
1606
Public Key Verifier
1607
Verification Output

Figure 16

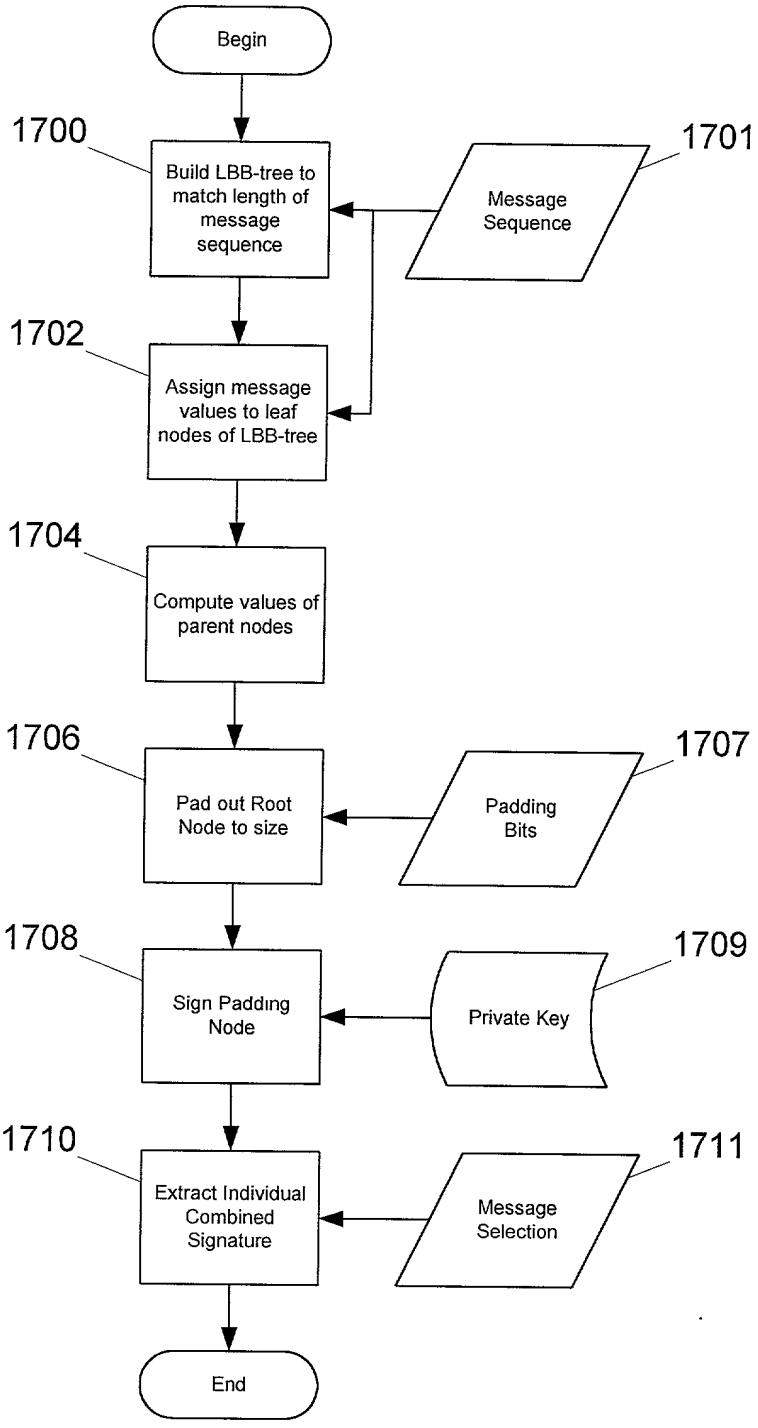

Begin
1700
Build LBB-tree to match length of message sequence
1701
Message Sequence
1702
Assign message values to leaf nodes of LBB-tree
1704
Compute values of parent nodes
1706
Pad out Root Node to size
1707
Padding Bits
1708
Sign Padding Node
1709
Private Key
1710
Extract Individual Combined Signature
1711
Message Selection
End

Figure 17

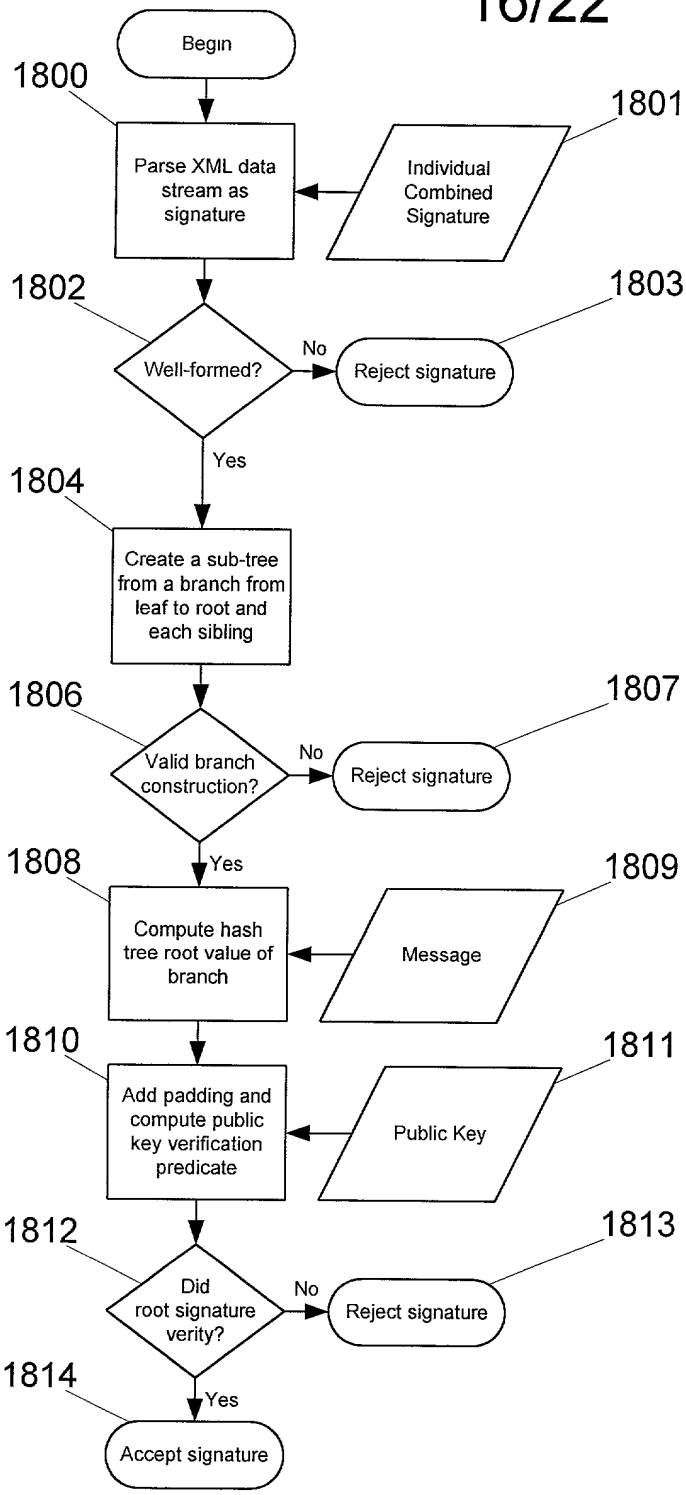

Begin
1800
Parse XML data stream as signature
1801
Individual Combined Signature
1802
Well-formed?
No
1803
Reject signature
Yes
1804
Create a sub-tree from a branch from leaf to root and each sibling
1806
Valid branch construction?
No
1807
Reject signature
Yes
1808
Compute hash tree root value of branch
1809
Message
1810
Add padding and compute public key verification predicate
1811
Public Key
1812
Did root signature verity?
No
1813
Reject signature
Yes
1814
Accept signature

Figure 18

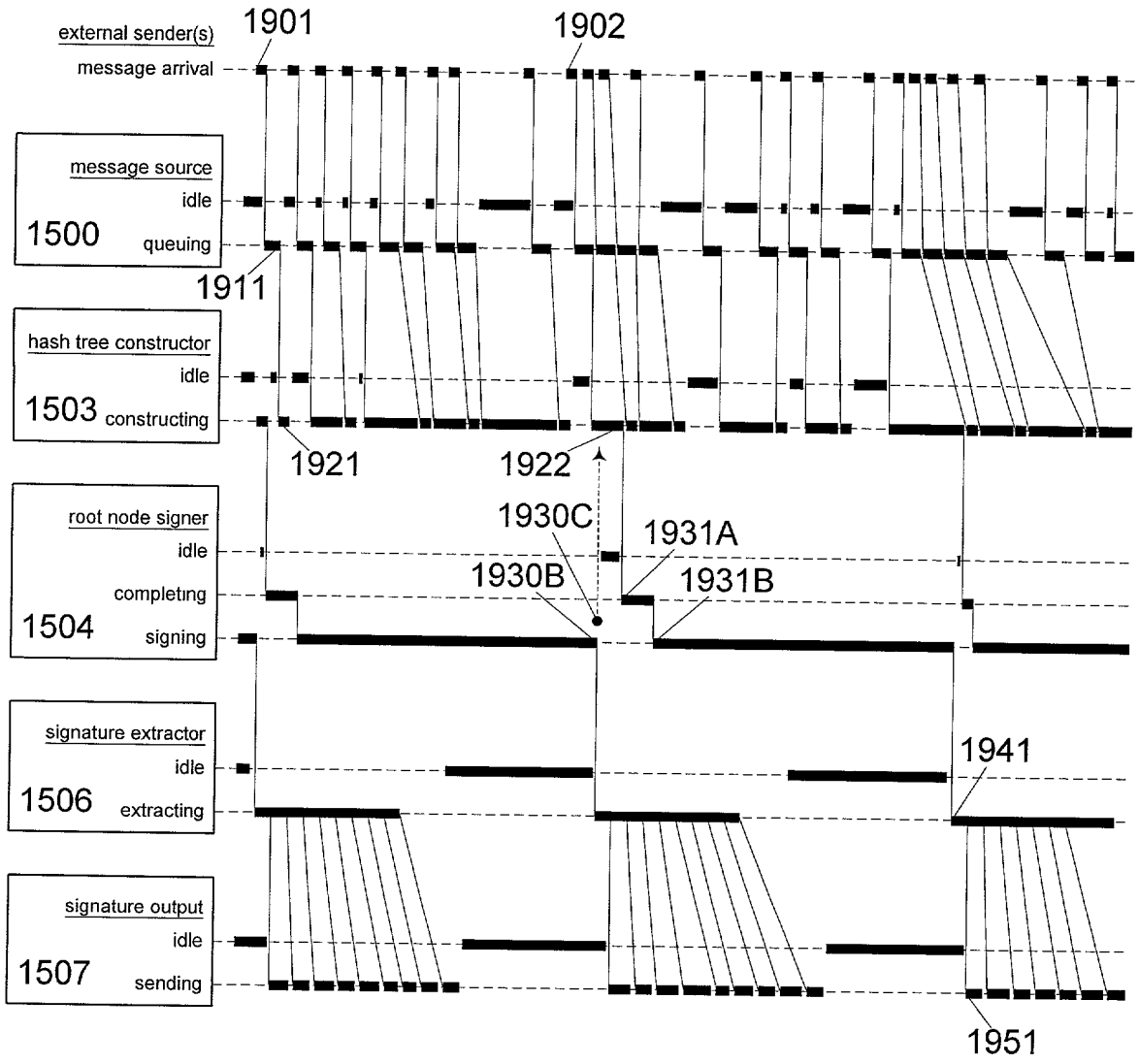

external sender(s)
message arrival
1901
1902
message source
idle
1500
queuing
1911
hash tree constructor
idle
1503
constructing
1921
1922
root node signer
idle
completing
1504
signing
1930C
1931A
1930B
1931B
signature extractor
idle
1506
extracting
1941
signature output
idle
1507
sending
1951

Figure 19

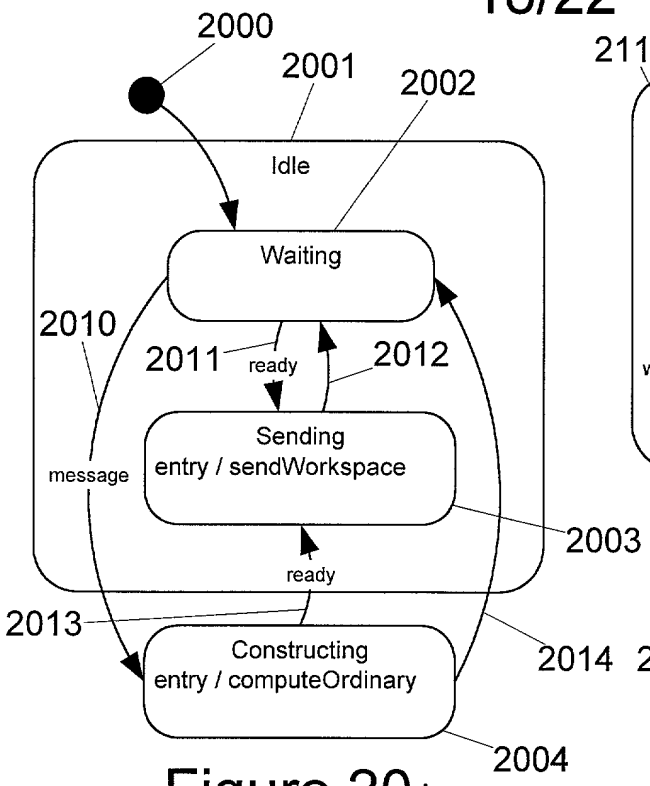


Figure 20

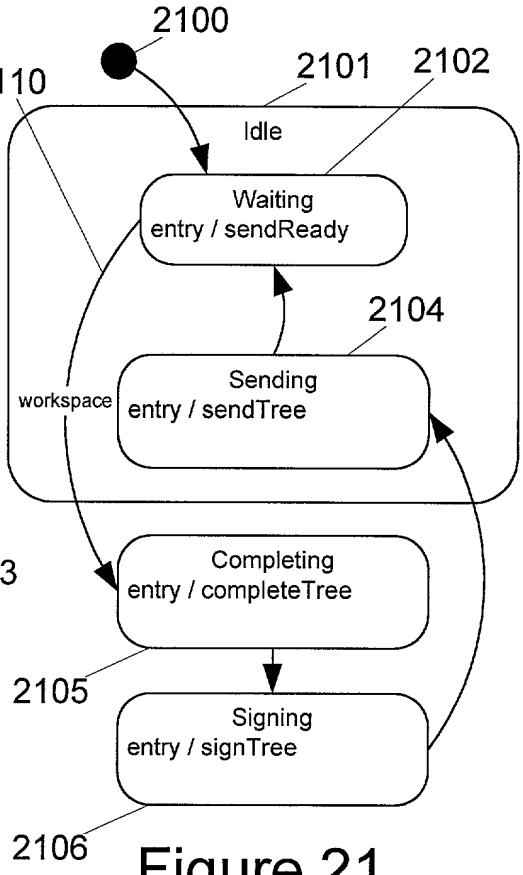


Figure 21

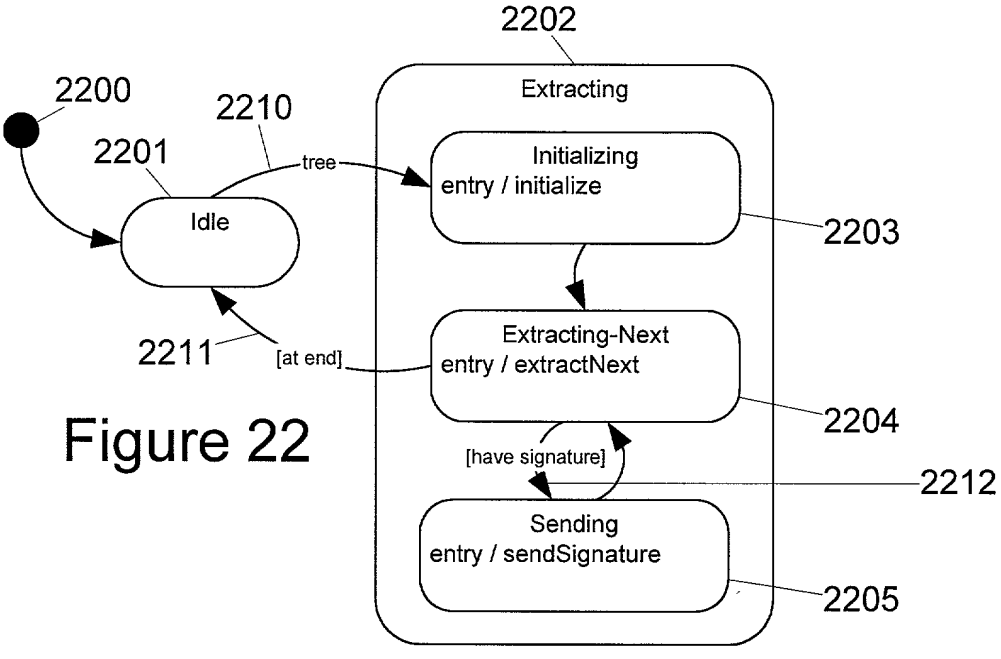


Figure 22

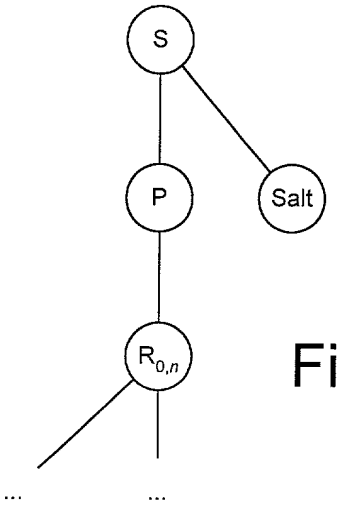


Figure 23A

23.7.3 $\text{NodeHash}_{n,salt} : \mathbf{H} \times \mathbf{H} \to \mathbf{H} : \langle x, y \rangle \mapsto F(salt + \text{NodeFormat}_n(x, y))$

23.13.5 $$\text{SignTagStart} : \mathbf{J} \times \mathbf{H} \to \Sigma^* : \langle j, salt \rangle \mapsto \begin{bmatrix} \text{"<signature"} \\ +\text{"value="}+\text{quote}+\text{Base64}(j)+\text{quote} \\ +\text{"salt="}+\text{quote}+\text{Base64}(salt)+\text{quote} \\ +\text{">"} \end{bmatrix}$$

Figure 23B

24.8.1 $$\text{NodeFormat}_n : (\langle i,j \rangle, x, y) \mapsto \begin{cases} \text{LayerFormat}(\text{"bottom"}, x, y) & \text{when } |i-j| = 1 \\ \text{LayerFormat}(\text{"top"}, x, y) & \text{otherwise} \end{cases}$$

24.1 $$\text{LayerFormat}(name, x, y) = \text{"<"} + name + \\ +\text{"left="}+\text{quote}+\text{Base64}(x)+\text{quote} \\ +\text{"right="}+\text{quote}+\text{Base64}(y)+\text{quote} \\ +\text{"/>"}$$

Figure 24

$$25.8.1 \quad \mathrm{NodeFormat}_n : (\langle i,j \rangle, x, y) \mapsto \left| \begin{array}{l} "< \mathrm{node}" + \\ + "\mathrm{pos} =" + \mathrm{Base10}(i) + "," + \mathrm{Base10}(j) \\ + "\mathrm{left} =" + \mathrm{quote} + \mathrm{Base64}(x) + \mathrm{quote} \\ + "\mathrm{right} =" + \mathrm{quote} + \mathrm{Base64}(y) + \mathrm{quote} \\ + "/>" \end{array} \right.$$

Figure 25

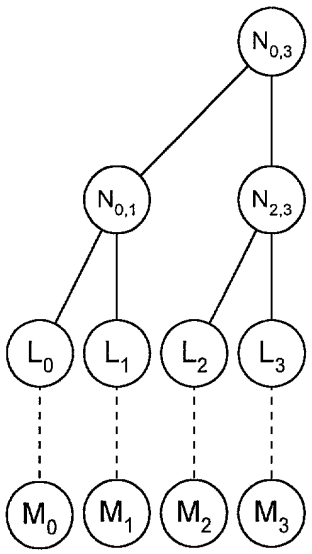


Figure 26A

$$26.7.6 \quad N_{\overline{M}} : t \mapsto \begin{cases} \mathrm{NodeHash}_{\#\overline{M}}(t, N_{\overline{M}} \circ \mathrm{Left}_n(t), N_{\overline{M}} \circ \mathrm{Right}_n(t)) & \text{when } t \in \mathbf{T}^+_{\#\overline{M}} \\ \mathrm{LeafHash}(M_t) & \text{when } t = \langle i,i \rangle \end{cases}$$

$$26.1 \quad \mathrm{LeafHash} : \mathbf{N} \times \mathbf{H} \to \mathbf{H} : x \mapsto F(\mathrm{LeafFormat}(x))$$

$$\mathrm{LeafFormat}(x) = "< \mathrm{leaf}"$$

$$26.2 \quad + "\mathrm{value} =" + \mathrm{quote} + \mathrm{Base64}(x) + \mathrm{quote}$$

$$+ "/>"$$

Figure 26B

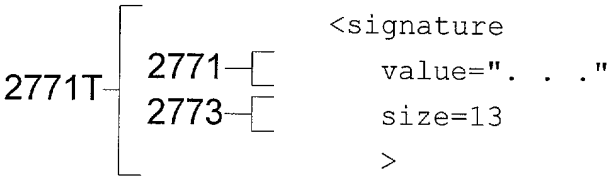
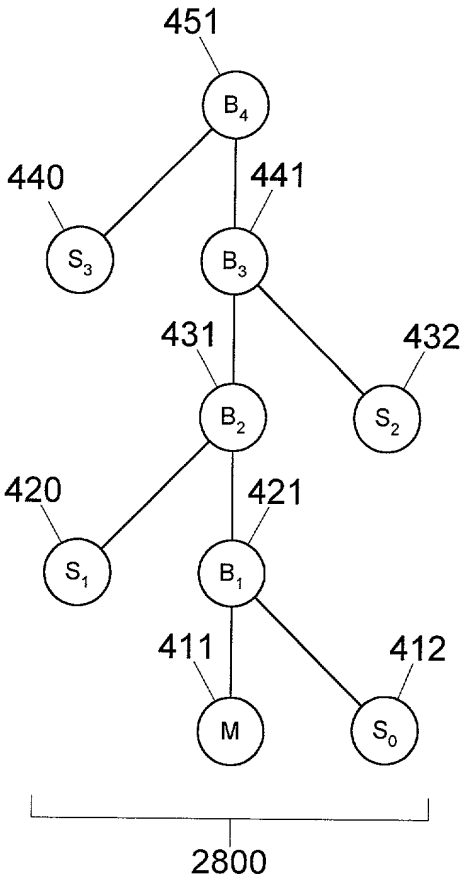


Figure 27



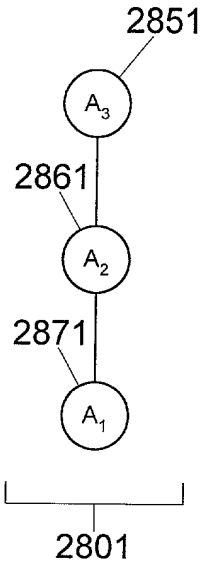


Figure 28

29.1 $\mathbf{W}'_L = \mathbf{W}_L \setminus P$: nodes of the base tree of a verification tree

29.2 $N'_{XS} = N_{XS}|_{\mathbf{W}'_L}$: verification tree

29.3 $V_{MR} : \mathbf{K}_D \times \mathbf{J} \rightarrow \mathbf{J}$: message recovery operation for signature verification

29.14.9 "signature is valid for $\langle d, M \rangle$" $\leftrightarrow$

$$\begin{cases} \text{"signature is well-formed"} \\ \wedge\ P_{XS}(B_0) \text{ is defined} \\ \wedge\ \exists i : P_{XS}(B_0) = \langle i, i \rangle \\ \wedge\ \text{PadStrip} \circ V_{MR}(d, sig) = N_{XS}(B_L) \end{cases}$$

Figure 29